

12-2009

Conceptualisation and Facilitation of SOA Governance

Christian Janiesch

SAP Research CEC Brisbane, c.janiesch@sap.com

Axel Korthaus

Faculty of Information Technology, Queensland University of Technology, axel.korthaus@qut.edu.au

Michael Rosemann

Faculty of Information Technology, Queensland University of Technology, m.rosemann@qut.edu.au

Follow this and additional works at: <http://aisel.aisnet.org/acis2009>

Recommended Citation

Janiesch, Christian; Korthaus, Axel; and Rosemann, Michael, "Conceptualisation and Facilitation of SOA Governance" (2009). *ACIS 2009 Proceedings*. 89.

<http://aisel.aisnet.org/acis2009/89>

This material is brought to you by the Australasian (ACIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in ACIS 2009 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Conceptualisation and Facilitation of SOA Governance

Christian Janiesch
SAP Research CEC Brisbane
SAP Australia Pty Ltd
South Brisbane, Queensland, Australia
Email: c.janiesch@sap.com

Axel Korthaus, Michael Rosemann
Faculty of Information Technology
Queensland University of Technology
Brisbane, Queensland, Australia
Email: {axel.korthaus|m.rosemann}@qut.edu.au

Abstract

As organisations strive to improve their capabilities in the areas of Service Management and Service-oriented Architectures (SOA), SOA Governance is becoming an increasingly important success factor. However, the concept of SOA Governance is complex and not well-understood, and the adoption of an adequate SOA Governance approach in an organisation can be difficult. Tools that support SOA Governance mostly have a technical bias and rarely address organisational aspects. In this paper, we contribute to the field by specifying a conceptual meta model for SOA Governance that integrates the structure of major IT and SOA Governance frameworks into one consolidated view. By presenting this conceptualisation and a corresponding prototypical implementation of a tool that supports SOA Governance maturity assessment, reference framework exploration and company-specific tailoring of SOA Governance, we provide insights into the first step of a Design Science research project, i.e. the development of an important IT artefact.

Keywords

SOA Governance, conceptual model, meta model, tool support.

INTRODUCTION

The paradigm of service-orientation and particularly the concept of Service-oriented Architectures (SOA) have been gaining considerable and increasing attention in most organisations for several years now. This can be contributed to the expected benefits not only from an IT perspective (e.g. reduced development and maintenance costs, faster IT response to business change, better quality through service reuse, etc.), but from a business perspective as well (e.g. business agility, reduced time to market, right-sized business model, etc.) (Marks 2008). As organisations strive to improve their capabilities in the areas of Service Management and SOA, SOA Governance is becoming an increasingly important success factor. Given that the lack of a comprehensive governance approach has been cited as the most common reason for failures of post-pilot SOA projects (Malinverno 2006), work in this area is highly relevant. As Marks (2008) puts it:

“SOA governance is mandatory for any measure of SOA success. Understanding and implementing effective SOA governance has become a corporate imperative.”

However, while its importance is without any doubt, there is a lack of a widely accepted terminological clarification. According to the Open Group (2009), appropriate SOA Governance has to define what decisions need to be made to have effective governance, who should make these decisions, how these decisions will be made and monitored, and what organisation structures, processes, and tools should be deployed in the organisation. Since there is no agreed-upon standard definition of SOA Governance in the literature, we provide a working definition of SOA Governance that is based on the definitions in Bernhardt and Seese (2008) and Dodani (2006): SOA Governance focuses on the decisions across the entire service life cycle to enable organisations to realise the benefits of an SOA. It is an approach to exercising control and mitigating risk by establishing organisational structures, processes, policies, and metrics suitable to ensure that the adoption, implementation, operation and evolution of an SOA is in line with the organisation's strategies and objectives and complies with laws, regulations and best practices.

The adoption and implementation of an appropriate SOA Governance approach can be a complex endeavour. Available reference governance models, such as COBIT (IT Governance Institute 2007) and ITIL (Office of

Governance Commerce 2007) as examples from the area of IT Governance can facilitate this process significantly and serve as a helpful starting point. They are, however, unable to provide detailed guidance on the role-activity-assignment and cannot facilitate individualisations that are required as different organisations with different levels of SOA maturity, sizes, existing governance structures, etc. will typically require customised approaches. The issue aggravates when multiple frameworks have to be deployed at different levels of the organisation. Thus, in order for organisations to be able to arrive at a focused and customised SOA Governance regimen, an approach is required that allows assessing their current state, selecting from potentially multiple related IT and SOA Governance frameworks, adjusting the components of these frameworks to organisation-specific needs, and then deriving an achievable roadmap for delivering the required or desired governance.

Such an SOA Governance framework adoption and implementation process needs to be facilitated by suitable tool support. Software tools for SOA Governance typically have a strong technical bias and target either design-time or runtime governance (see Section 2). While those perspectives and toolsets are valid and required for any effective SOA initiatives that go beyond an initial pilot project, tool support for the introduction of organisational aspects of SOA Governance is less prevalent.

“SOA governance has been co-opted by technologists to some extent, and this has been a disservice, leading to an over-focus on tools and technology standards and not enough emphasis on the processes and organizational models of governance” (Marks 2008).

Against this background, the design science approach chosen in this paper is to create a suitable theoretical underpinning for the conceptualisation of a tool, a model, that focuses on supporting the process suggested above. It has to allow for a multi-level, multi-framework customisation of major IT and SOA Governance frameworks to derive a tailored SOA Governance framework for organisational aspects of SOA Governance. We develop and describe a generic governance meta model that integrates the structure of these existing frameworks into one view.

The remainder of this paper is structured as follows. First, we review the related body of knowledge regarding IT Governance, SOA Governance, the conceptual modelling of governance frameworks and current software support for SOA Governance. The first main part of the paper will then present in detail our proposal for a governance meta model, that is derived from existing IT Governance frameworks and empirical insights gained from interviews with consultants from a large ERP vendor, an Australian government agency, and a major Australian retailing company. The second main part of the paper is dedicated to the topic of tool support for SOA Governance and describes the conceptualisation and a first prototype of a software application that can provide support for the organisational aspects of SOA Governance. The paper concludes with a summary and outlook.

RELATED WORK

Numerous IT Governance frameworks exist such as COBIT, ITIL, ValIT, ISO 20000, ISO 17799, etc. However, each of them focuses on a specific aspect of a company's IT governance. While the IT Infrastructure Library (ITIL), for example, mainly deals with management and support process definitions (Office of Governance Commerce 2007), the ISO 17799 standard primarily targets security management (International Organization for Standardization 2006). When these approaches are juxtaposed, they do not exclude but rather complement each other. In comparison, COBIT is a high level governance and control framework, more tightly aligned with the business objectives of an organisation than with operational issues (IT Governance Institute 2007). As a matter of fact, most frameworks somehow align with COBIT. It has, so far, become a de facto standard for IT control globally, and its implementation increasingly gains interest.

Concerning SOA and Service Governance, many software companies introduced their own definitions in white papers. A large number of different approaches have been proposed which cater for specific aspects of governance concerning an SOA. For example, some are limited to change management aspects of an SOA while others do not even include roles and accountabilities and predominantly cover service design. In summary, they lack framework scope and are often driven by own market interests (Janiesch et al. 2009; Niemann et al. 2008).

The most promising development with regard to an emerging SOA Governance Framework is the work conducted by the Open Group (The Open Group 2009). Their proposal for a SOA Governance framework describes the governance activities that are impacted by an SOA and puts forward some best-practice governance rules and procedures for those activities. However, as of September 2009 it has only the status of a Draft Technical Standard and still lacks essential elements such as a specification of detailed accountabilities along the service life cycle or a sound ontology that related the core governance elements to each other.

Related to our goal to provide a governance meta model is the work of Bernhardt and Seese (2008) who propose a conceptual SOA Governance framework striving to cover the complete SOA life cycle. Their approach uses the standardised OASIS SOA reference model (MacKenzie et al. 2006) as a starting point for the identification of SOA Governance aspects. The model, however, is not grounded in empirically tested practises as documented in the related IT Governance literature. Bernhardt and Seese (2008) have not yet investigated the relationships between their approach and common IT frameworks.

Similar to the approach taken in this paper, Goeken and Alter (2008) promote the use of meta models to represent IT Governance frameworks. They use the widespread COBIT Governance framework as an example to show how a conceptual meta model can be used to establish a theoretical foundation, formalise such frameworks and provide a means for analysing, comparing, and relating them to each other. The authors also point out the positive side effect of meta modelling as an activity that can provide a starting point for the representation of governance frameworks in application systems and tools. Their aim is to compare the models of individual frameworks and allow their merge rather than attempting to unify their content into a comprehensive and integrated meta model for SOA Governance.

Most of the software tools that claim to support SOA Governance have a strong technical focus (e.g., what activities are required when a server is unavailable), often targeting support for either design-time or runtime governance as part of service development life cycle governance (Marks 2008). Service development life cycle governance refers to governing the whole life cycle of services from the service idea/ proposal through requirements, analysis, design, implementation, quality assurance, and testing to registration/ publishing, aggregation, orchestration, provisioning, consumption, and retirement. Design-time governance comprises only the first part of this life cycle and emphasises compliance to architecture and technical design standards. Runtime governance starts with the event of an implemented and tested service being registered/ published and emphasises operational requirements for performance, quality of service service-level agreements, and security (Marks 2008).

According to Marks (2008), platforms for design-time governance and runtime governance can be distinguished by their characteristics. The former typically guide production and consumption of services from initial inception to selection for use in end-user applications and involve many different types of assets beyond services and schemas, such as components, legacy APIs, design patterns, etc. Information must be rendered and presented through different channels, such as browsers, integrated development environments, reports, etc. Depending on the preferences of the target role, and integration with heterogeneous development tools, such as supply chain management systems, document management systems, quality assurance and test systems, defect tracking systems, etc. is required. Runtime governance platforms, on the other hand, are complementary in the sense that they typically require minimal end user interaction, high throughput and responsiveness and relate to the behaviour of a deployed service when called, the enforcement of various policies, the validation of behaviour and the replacement and retirement of services. The more limited set of assets they are focused on (as opposed to design-time governance platforms) includes service interfaces/ implementations and deployment policy configurations.

As argued above, a clear gap in related work is the lack of a comprehensive and unified view on SOA Governance as most of the work so far has been patchwork. The framework we propose is primarily derived from the much wider area of successful IT Governance frameworks in order to leverage existing knowledge and revise it against the background of SOA-specific characteristics. Concerning tool support, the lack of coverage of organisational issues indicates a clear gap as most tools have a technical focus. We aim at closing this gap with the tool structure we propose.

A META-MODEL-BASED VIEW ON SOA GOVERNANCE

Overview

A main objective of this paper is to unify the structure of major IT and SOA governance frameworks into one view on governance. As argued above, most existing frameworks or approaches focus on a specific aspect of governance. Thus, it is important to first create a consolidating overview of the constituent parts and their interrelationships. Conceptual models are artefacts which are expressed using a conceptual modelling method (Wand and Weber 2002). Similarly, governance frameworks can be conceptually modelled (Goeken and Alter 2008). For the semi-formal conceptualisation of SOA Governance we propose the use of a meta model. A meta model puts as a 'model of a model' (Oei et al. 1992; Strahinger 1996) all constructs of a domain in context within an integrated model that differentiates basic semantic relationships. When analysing different governance frameworks, we identified three major areas of concern:

- A. Processes, roles and other essential attributes

B. Views and indicators

C. Company-specific information

These three clusters are linked to each other. Part A comprises the essential constructs of any governance framework. Part B comprises the non-essential attributes which add value to the framework but are not indispensable for an operationalised SOA governance model. Part C is not part of any framework as such but provides the necessary alignment to the overall corporate governance. It is important to relate a governance framework blueprint to company facts in order to properly operationalise it. Figure 1 gives a summary of the above.

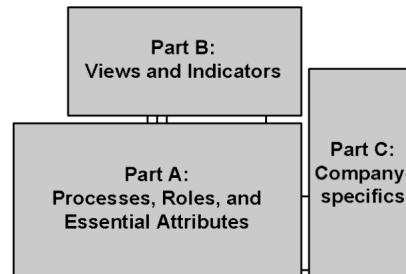


Figure 1: Clusters of the Governance Framework

Part A – Processes, Roles, and Essential Attributes

Basically all governance frameworks centre on the corporate processes. *Processes*, also termed tasks or activities, in governance frameworks de-/ prescribe procedures to ensure the desired operation of an organisation (and its IT or SOA in particular). Processes consist of subordinated processes which in turn might be composed of sub-processes.

Each process is linked to a number of *roles* which use and create *artefacts* as inputs and outputs of the performed tasks. Weill and Ross (2004) stress the centrality of stakeholders in their definition of IT Governance. Consequently, it is necessary to unambiguously define roles as well as their rights and competencies. Weill and Ross suggest specifying a decision rights and accountability framework. RACI matrices are a common method to clarify such responsibilities (cf. Section 4 for more details on our use of RACI). The same holds true for the artefacts mentioned before. Each artefact requires exactly one role responsible as well as at least one role accountable. Furthermore, the (read, write, delete) access to existing artefacts might be restricted to certain roles. Artefacts comprise all sorts of documents, such as status updates, roadmaps or architecture diagrams. Artefacts in a wider sense include further items such as data or code.

Each process has *key performance indicators (KPI)* or *goals*. These are calculated or derived from *metrics* that are collected through monitoring. KPIs as well as goals can be composed of different individual objectives. For example, to measure the effective use of assets, the return on investment ratio can be used. It is based on the net income and totals assets which are two individual ratios (Groppelli and Nikbakht 2006). For each ratio there is at least one role responsible. Roles can also have targets which are not directly connected to a governance process.

Processes are executed using *tools*. These can be as simple as pen and paper but nowadays more commonly involve software. Usually, the tool is used to convert the input artefact to the output artefact which is handed over to the subsequent process. We acknowledge the fact that access to tools, especially software, might require a similar access management as artefacts do. For the sake of clarity we did not include this relationship in the meta model below.

Each process requires certain *skills* to be properly executed. For example, when designing IT architectures it is important that the role knows or is able to use the involved methods such as conceptual modelling as well as the software tools such as an SAP system. There are different ways to assign skills. It can be attached to the job description of a role. However, when the involved processes and tools change, the job description has to be kept in sync. On the other hand, the required skills of a role can simply be regarded as the sum of a role's associated process requirements as well as their tool requirements. If, however, processes and tools change, the skill requirements of a role change, too. Over time, this can create unbalanced role definitions. As it is also a good means to evaluate the balance of a framework's roles, we made the conscious decision to attach required skills to tools and processes, not to roles themselves.

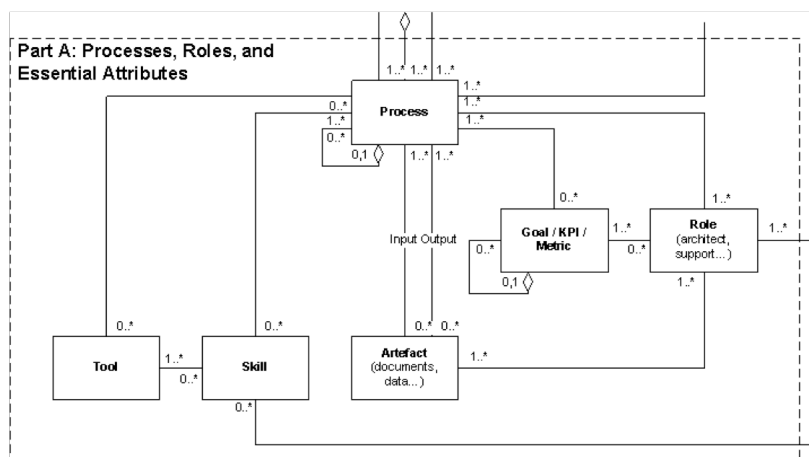


Figure 2: Processes, Roles, and Essential Attributes

Part B – Views and Indicators

As part A revealed, processes can be regarded as the core of a governance framework. There are usually a large number of processes to consider. Thus, it is important to classify and structure processes in order to make them manageable. Most commonly, processes are grouped in *phases*. Usually, the phases are aligned with the life cycle of the governance framework's objectives (e.g. phases for planning, implementation, delivery, and monitoring for IT governance). Thus, phases can be regarded as the primary structuring characteristic to create views on a framework. Consequently, the governance *model* as such (e.g. COBIT, ITIL) is related to its processes through phases.

Processes are related to governance *focus areas* which describe the topics that executive management needs to address within their enterprises. Focus areas provide a means to highlight the processes relevant for a certain topic. Examples for focus areas are strategic alignment, value delivery, resource management, risk management, and performance management (IT Governance Institute 2007). Focus areas can also be understood as organisational domains of a governance framework. Respective examples are strategic alignment, business process management, enterprise service management, enterprise service controlling (SAP AG 2007). Focus areas could also be used to classify roles or artefacts.

Similarly, *management layers* can be used to attribute processes to different organisational entities such as strategy management, portfolio management, programme management, project management, and operations management (Office of Governance Commerce 2007). Ultimately, due to conceptual similarities, management layers and focus areas could be subsumed to a construct named *process area* or process domain with an associated construct *process area type*.

SOA Governance typically evolves over time and grows in maturity the more the organisation develops SOA capabilities. Most maturity models distinguish five *maturity levels* (initial, repeatable but intuitive, defined, managed and measurable, optimised) (IT Governance Institute 2007; Software Engineering Institute 2009). Maturity models are a means to measure the maturity of a system concerning a structured set of aspects. Each process is either associated with one *maturity level* or each process has one occurrence for every maturity level available in the governance framework. An example for the former is a framework which prescribes a process such as requirements management already at a low level (repeatable) while it recommends another process such as risk management for a more advanced maturity level (defined). An example for the latter is the execution of a process in varying level of detail. The process to define a strategic IT plan, e.g., can be classified from initial, if it is executed ad hoc, to optimised, if it is a documented, living process that is continuously performed. It is possible to combine these two approaches and classify processes into a specific maturity level while at the same time assessing their status of implementation. I.e., the process of risk management might be classified as defined, but if it is only executed in an ad hoc fashion, its state of implementation must be considered as initial. *Maturity level models* are used to distinguish different maturity models used (for an overview of maturity models cf. e.g. Becker et al. 2009). Capability profiles may be linked to maturity models.

They represent the application of the model on a system and outline the overall abilities of the system compared with the planned targets.

Figure 3 summarises the above. The constructs *phase*, *maturity levels*, *management layers*, and *focus areas* are connected to *process* (cf. Part A).

Part C – Company-specific Information

In order to apply a governance framework to an individual company, governance roles need to be matched to company-internal structures. Simplified, a *company* consists of several *organisational units* in which several *positions* are available. An *employee* of the company, an individual *person*, holds one of these positions. Each position can correspond to one or multiple roles in the governance framework. Accordingly, an employee is involved in the execution of governance processes in one or even multiple roles.

In order to ensure the flawless execution of processes, the skills of each employee should be documented in the company's human resources (HR) system. Ideally, his skills are a superset or close match to the skills required by his role(s).

Apart from this data, it is conceivable to also document information concerning technological capabilities or company-specific information on existing processes for a framework matching. At this point, we abstract from that.

Figure 4 summarises the above. The construct *position* is connected to *role* (cf. Part A) while the construct *person/employee* is connected to *skill*.

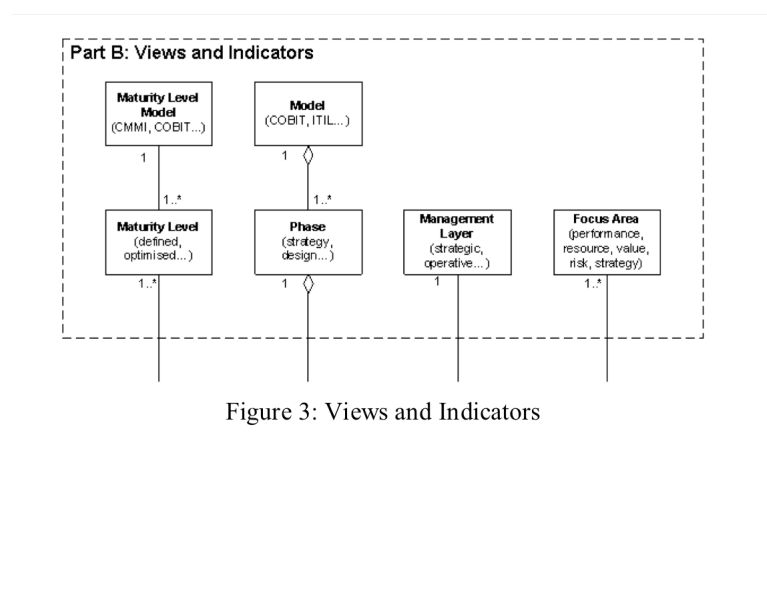


Figure 3: Views and Indicators

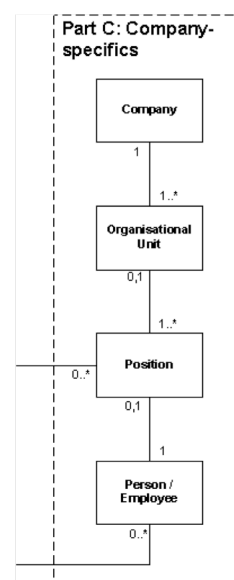


Figure 4: Company-specific Information

Integrated View

Figure 5 provides an integrated view on the above parts. While part A comprises the essential constructs of any governance framework, part B includes constructs to further classify governance processes. Part C includes company-specific information which must be included when operationalising the framework. Also, we have to acknowledge that in principle every construct should be recursively applicable. I.e., a tool can be composed out of different individual tools as well as one layer of phases might not be sufficient. For the sake of simplicity we did not include this requirement in the meta model. We only highlighted those constructs explicitly which have to be nestable.

These three parts completely cover our proposed meta model for SOA Governance. The resulting meta model has undergone an evolution based on the consideration and integration of first empirical insights gained through the validation of an earlier version of the meta model in six interviews with SAP consultants in Germany as well as Asia Pacific, two two-day workshops with a Western Australian government agency, and a phone interview with the Manager, Competency Centres Design, of a major Australian retailer. The first

version of the meta model, for example, did not include capturing accountabilities of roles for the creation of particular artefacts, but was limited to processes, and also refrained from modelling company-specific organisational information. The analysis of the approach used at the retailing company, for example, led to the addition of the construct *artefact* to the model and pointed out the relevance of providing a possibility to link roles etc. to concrete positions and employees in the organisation. Examples of artefacts that needed to be governed at the retailing company included SOA Roadmaps, SOA KPI reports, business cases, service candidates, service models, test plans, and many more. Interviews with SAP consultants, on the other hand, revealed that skills management, roadmap generation, and benchmarking against best practices are crucial aspects that were not sufficiently accounted for in the first version of the meta model.

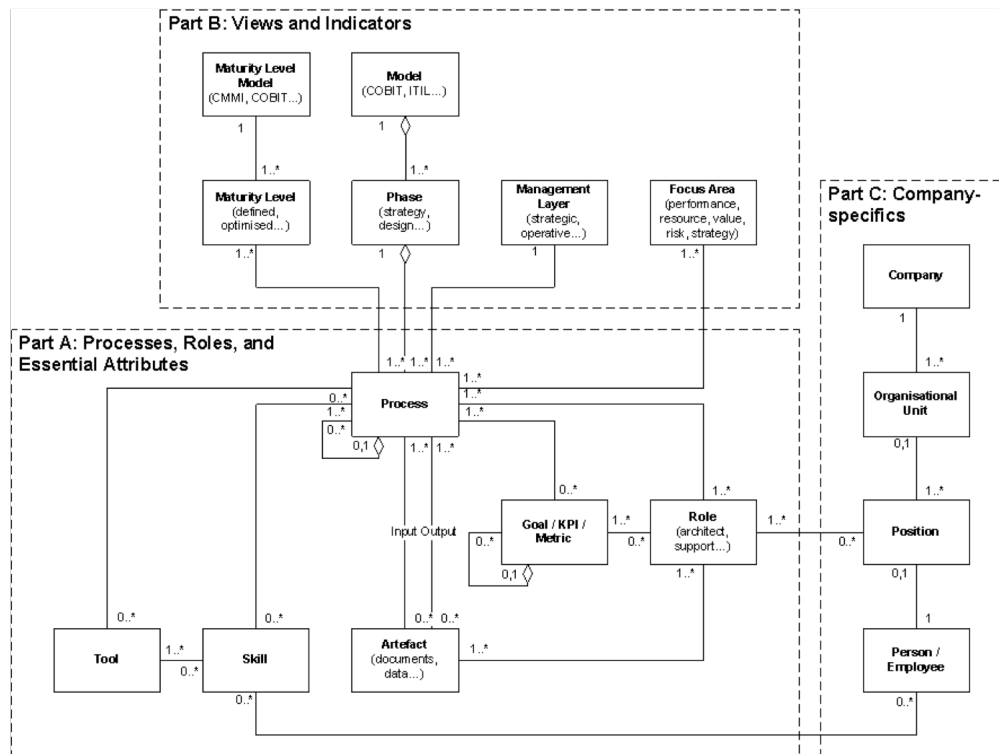


Figure 5: Integrated View on Governance

TOOL SUPPORT FOR SOA GOVERNANCE

Conceptualisation of a tool for managing organisational aspects of SOA Governance

While a meta model for SOA Governance can provide a sound integrated conceptualisation, it itself cannot easily be converted into an operational SOA Governance. In particular, it needs to be populated, i.e. for example a set of roles has to be provided to give organisations detailed guidance. In the following, we report on the conceptualisation and partial implementation of a tool, which is intended to help external as well as internal consultants or training professionals deliver effective SOA Governance consulting to their internal or external clients. It can equally well be used by internal SOA Governance experts to support their tasks. The functional architecture of the tool is derived from the elements specified in the SOA Governance meta model introduced above. The implementation as a web application allows the use of the tool as both a standalone offline application on a desktop or laptop computer (requiring a running local web server) and an online application that could be hosted, e.g., by a consultancy and accessed via the Internet in client companies or in a large organisation to be accessed in their Intranet.

The envisioned software tool consists of three major components:

- the *Governance Framework Analyser* component,
- the *Governance Framework Builder* component, and
- the *Governance Framework Manager* component.

In the following, we first give a brief overview of all three components of the tool. We emphasise the Governance Framework Manager component in more detail, as it has already been implemented as a prototype.

The *Governance Framework Analyser* component provides basic functionality to help the organisation capture the status quo of its organisational structure and its maturity level. To this end, it offers the following features: First of all, the Analyser allows the end user to *model the company* by entering data related to the company's organisational units, positions, the employees that hold these positions and general information including benchmark data such as the ratio of IT staff per 100 employees. This feature basically implements Part C of the meta model presented in Figure 4. Moreover, a user is able to enter descriptive information about the (service-related) processes that are currently in place in the company. After having selected an existing governance reference framework (e.g. COBIT or ITIL) from the extensible set of frameworks that come with the tool, the core feature is the possibility to map roles and processes from the reference governance framework to roles and processes that exist in the company. A link is established from company-specific positions and processes to the available roles and processes in Part A of the meta model. For example, a framework might use the term *chief architect* for a proposed role, which should be mapped to the company-specific *enterprise architect* role in a particular user company because this appears to be the most similar counterpart to the framework role. A very important constituent of the Analyser component is the assessment of the current level of maturity the user company has achieved with regard to its SOA initiative. This information refers to Part B of our meta model. In this part of the Analyser component, each process of the user company can be assigned a value that indicates the maturity level of that process. This information will then be particularly useful in the context of the last feature of the Analyser component, which reflects in the form of a radar (spider) diagram the overall SOA maturity of the company in relation to the targeted process maturity levels required or recommended by the reference governance framework. Also, it is possible to compare the required set of roles of the framework processes to the existing positions in the company as well as to match the skills required by the processes and used tools to the actual skills documented for the company's employees. Based on this juxtaposition, strategic directions for advancing the adoption and implementation of the targeted governance framework can be derived and further steps can be agreed upon. This tool can by no means be a solution to depict an organisation up to a 100 % as the innate nature of SOA is change. However, we aim at providing an 80:20 solution to be able to highlight problem areas and prioritize change.

The second component of the tool is the *Governance Framework Builder* component. The core idea of this component is to enable a user company to mix and match constituent parts of different reference governance frameworks in order to derive its own governance framework. This component functions like an intelligent *construction kit* that also provides several consistency checks which aim at avoiding the composition of user-specific frameworks that have logical inconsistencies or other inherent problems due to the combination of incompatible parts from different reference frameworks. Examples for inconsistencies include but are not limited to omitting required procedures of included processes, the deployment of equivalent processes or roles from different frameworks or the disregard of separation of power of roles. Further steps include a preliminary evaluation of costs, and roadmapping capability.

The third component of the envisaged tool is the *Governance Framework Manager*, which allows the user company-specific adaptation and management of the governance model. The prototype emphasises the organisational aspects of SOA Governance by utilising RACI matrices as mentioned earlier. The RACI approach is a key best practice that organisations should adopt to maximise the effectiveness of their SOA Governance efforts. The prototype comes with a proposed reference SOA Governance framework that allows the assignment of accountabilities based on a detailed mapping of the involvement of the different roles in the activities of a SOA life cycle that was derived from literature reviews, the analysis of job descriptions and interviews of case study partners. RACI charts are specified for each of three management layers (service portfolio management, service project management, and service operations management) to show the recommended accountabilities. The RACI charts map activities of the SOA life cycle to roles of stakeholders in a SOA initiative and propose their responsibilities by specifying which roles are (r)esponsible (i.e., perform the actual activity), (a)ccountable (i.e., ensure that the activity happens), (c)onsulted (i.e., should be consulted prior to decision or action) or (i)nformed (i.e., need to be informed of the decision or action after it is made) regarding specific activities. Roles are represented as columns and SOA life cycle activities as rows. By providing these RACI charts, the reference framework offers a practical tool for the analysis of responsibilities along the whole SOA life cycle.

According to Morgan (2009), developing RACI charts is typically a political process and surfaces many organisational issues because it confronts role conceptions, role expectations and role behaviours. Morgan (2009) promotes RACI charting to counteract problems such as unclear accountabilities between individuals and departments, redundancies or work not being accomplished, delayed or incomplete work, inadequate communication and/ or co-ordination, unclear approval/ decision-making processes, multiple decision points in

a process, tasks being performed and decisions being made at the wrong levels, infighting for promotions or important assignments, finger-pointing when something fails, and the inability to manage the interfaces. Founding the organisational aspects of a SOA Governance approach on RACI matrices has the potential to help clarify individual expectations, reduce redundancy in tasks, improve communication, reduce levels of approval in processes, eliminate consolidators and coordinators and define interfaces between individuals.

Managing these RACI charts as its core artefact, the Governance Framework Manager component offers the following five functions: reference model exploration, reference model tailoring, user model exploration, comparison of reference and user model, and reporting.

Reference model exploration refers to a feature that allows browsing the different roles specified by the reference model, accessing definitions of these roles, going through the different phases and management layers and their definitions, viewing the activities (processes) of the SOA life cycle and their descriptions, and most importantly analysing the assignment of accountabilities of roles for different activities. Reference model tailoring denotes the feature that allows a user to reduce or modify the reference model in order to adapt it to company-specific needs and thus to derive a user model. To this end, labels, such as the names used for roles or processes etc. can be edited and changed, but also deleted in case they are not needed. The deletion of parts of the reference RACI model of course necessitates a horizontal and vertical review process which ensures, for example, that each activity still has exactly one role that is accountable for its execution, or that the accountabilities a deleted role had possessed are now distributed to the remaining roles in the user framework. To support this process, the tool implements consistency rules which are checked automatically and lead to user notifications if the rules are violated. User model exploration is analogous to the first feature, this time, however, relating to the tailored version of the reference model, the user model. Comparison is a feature that reports on the differences between the reference governance model and the company-specific user model. This feature produces a report that clearly points out which elements have been renamed or deleted and how accountabilities have been reallocated. The last feature of this component is Reporting, which enables the user to query information on specific topics such as the processes accountable by a role, a comprehensive guideline for all artefacts used in a process, etc.

CONCLUSION

In this paper, we have argued that SOA Governance is a crucial element for organisations engaging in SOA initiatives to ensure that all SOA-related efforts jointly meet the requirements of the organisation and its stakeholders. We have pointed out that most SOA Governance tools have a technical focus and neglect important organisational aspects of SOA Governance. To be able to design and implement a tool that fills this gap and supports organisations in assessing their SOA maturity, perform benchmarking against best practices, learn from existing governance frameworks as well as customise or build an organisation-specific SOA Governance framework, we have developed and discussed a consolidated SOA Governance meta model that was informed by existing standard frameworks as well as empirical insights from interviews with practitioners. The meta model consists of three clusters covering processes, roles and other essential attributes, views and indicators, and company-specific information. Owing to its high level of conceptual abstraction and its rooting in well-proven IT Governance approaches, it is even generic enough to be not restricted to the particular perspective of SOA Governance.

We have also presented the conceptualisation of the envisaged tool. We discussed different areas of application for the software with the focus on analysis, creation and management of an SOA Governance approach. We described the first prototype that has been developed in detail and highlighted the importance of managing organisational aspects through RACI charts. Future work will include further validation of the meta model and subsequently the full implementation of the software for evaluation in practice.

REFERENCES

- Becker, J., Knackstedt, R., and Pöppelbuß, J. 2009. "Developing Maturity Models for IT Management: A Procedure Model and its Application," *Business & Information Systems Engineering* (1:3), pp 213-222.
- Bernhardt, J. and Seese, D. 2008. "A Conceptual Framework for the Governance of Service-Oriented Architectures," *Proceedings of the 3rd ICSOC Workshop on Trends in Enterprise Architecture Research (TEAR). Lecture Notes in Computer Science Vol. 5472*, Feuerlicht, G. and Lamersdorf, W. (eds.), Sydney, pp 327-338.
- Dodani, M.H. 2006. "Who Took the Cookie from the Cookie Jar?," *Journal of Object Technology* (5:4), pp 23-28.
- Goeken, M. and Alter, S. 2008. "Representing IT Governance Frameworks as Metamodels," *Proceedings of the 2008 International Conference on E-Learning, E-Business, Enterprise Information Systems, and E-Government (EEE)*, Las Vegas, NV, pp 48-54.

- Groppelli, A.A. and Nikbakht, E. 2006. *Finance*. 5th edition, Hauppauge, NY: Barron's.
- International Organization for Standardization. 2006. "ISO 17799 Central: The A-Z Guide for ISO 27001 and ISO17799/ISO27002" Retrieved 2009-03-16, from <http://www.17799central.com/>.
- IT Governance Institute. 2007. *COBIT 4.1: Control Objectives for Information and Related Technology*, Rolling Meadows, IL: IT Governance Institute.
- Janiesch, C., Niemann, M., and Repp, N. 2009. "Towards a Service Governance Framework for the Internet of Services," *Proceedings of the 17th European Conference on Information Systems (ECIS)*, Verona, pp 1-13.
- MacKenzie, C.M., Laskey, K., McCabe, F., Brown, P.F., and Metz, R. 2006. "Reference Model for Service Oriented Architecture 1.0. OASIS Standard." Retrieved 2009-07-09, from <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.html>.
- Malinverno, P. 2006. "Service-Oriented Architecture Craves Governance. Gartner Research." Retrieved 2009-07-09, from <http://www.gartner.com/DisplayDocument?id=488180>.
- Marks, E.A. 2008. *Service-Oriented Architecture Governance for the Services Driven Enterprise*, Hoboken, NJ: John Wiley & Sons, Inc.
- Morgan, R. 2009. "How to do a RACI Charting and Analysis: A Practical Guide." Retrieved 2009-07-09, from <http://www.projectsart.co.uk/pdf/how-to-do-raci-charting-and-analysis.pdf>.
- Niemann, M., Eckert, J., Repp, N., and Steinmetz, R. 2008. "Towards a Generic Governance Model for Service-oriented Architectures," *Proceedings of the 14th Americas Conference on Information Systems (AMCIS)*, Toronto, pp 1-10.
- Oei, J.L.H., van Hemmen, J.G.T., Falkenberg, E.D., and Brinkkemper, S. 1992. "The Meta Model Hierarchy: A Framework for Information Systems Concepts and Techniques," in *Technical Report Vol. 92-17*, University of Nijmegen.
- Office of Governance Commerce. 2007. *ITIL v3: Information Technology Infrastructure Library Version 3 Core OGC Titles Vol. 1-5*, London: The Stationery Office.
- SAP AG. 2007. "Enterprise Service-oriented Architecture Governance: Ensuring the Success of Enterprise SOA Projects. SAP Whitepaper," Walldorf.
- Software Engineering Institute. 2009. "CMMI Models and Reports." Retrieved 2009-07-08, from <http://www.sei.cmu.edu/cmmi/models/>.
- Strahringer, S. 1996. *Metamodellierung als Instrument des Methodenvergleichs: Eine Evaluierung am Beispiel objektorientierter Analysemethoden*. Dissertation, University of Darmstadt, Aachen: Shaker.
- The Open Group. 2009. "SOA Governance Framework. Draft Technical Standard v2.4." Retrieved 2009-07-09, from https://www.opengroup.org/projects/soa-governance/uploads/40/19263/SOA_Governance_Architecture_v2.4.pdf.
- Wand, Y. and Weber, R. 2002. "Research Commentary: Information Systems and Conceptual Modeling: A Research Agenda," *Information Systems Research* (13:4), pp 363-376.
- Weill, P. and Ross, J.W. 2004. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Boston, MA: Harvard Business School Press.

ACKNOWLEDGEMENTS

The project was funded by means of the German Federal Ministry of Economy and Technology under the promotional reference "01MQ07012". The authors take the responsibility for the contents.

Parts of this research have been funded by a research project within the Australian Research Council (ARC) Linkage Schema (grant code LP0669244) including financial support from SAP Research and the Queensland Government.

COPYRIGHT

Janiesch, Korthaus, Rosemann © 2009. The authors assign to ACIS and educational and non-profit institutions a non-exclusive licence to use this document for personal use and in courses of instruction provided that the article is used in full and this copyright statement is reproduced. The authors also grant a non-exclusive licence to ACIS to publish this document in full in the Conference Papers and Proceedings. Those documents may be published on the World Wide Web, CD-ROM, in printed form, and on mirror sites on the World Wide Web. Any other usage is prohibited without the express permission of the authors.